Automatic High Dynamic Range Image Generation for Dynamic Scenes

Katrien Jacobs[1], Celine Loscos[1,2], and Greg Ward[3]

keywords: High Dynamic Range Imaging

***Abstract*— Conventional High Dynamic Range Image (HDRI) generation methods using multiple exposures require a static scene throughout the low dynamic range image (LDRI) capture. This strongly limits the application field of HDRI generation. The system presented in this paper offers an automatic HDRI generation from LDRIs showing a dynamic environment and captured with a hand-held camera.**

The method presented consists of two modules that can be fitted into the current HDRI generation methodology. The first module performs LDRI alignment. The second module removes the ghosting effects in the final HDRI created by the moving objects. This movement removal process is unique in that it does not require the camera curve to detect movement and is independent from the .contrast between the background and moving objects. More specifically, the movement detector uses the difference in local entropy between different LDRIs as an indicator for movement in a sequence.

***Index Terms*— High Dynamic Range Imaging**

I. INTRODUCTION

Application domains such as image-based rendering and mixed-reality use photogrammetry when performing relighting and require input directly from photographs [5][7]. Photographs often present a loss of colour information since clipping and noise will occur in areas which are under- or over-exposed. This loss of information can have a crucial impact on the accuracy of the photogrammetric results. Methods have been created to combine information acquired from (low dynamic range) images (LDRIs) captured with varying exposure settings, creating a new photograph with a higher range of colour information called a high dynamic range image (HDRI). It has now become viable to use HDRIs in photogrammetry, and new cameras [2][15][16] are being developed with a higher dynamic range than the conventional cameras.

The drawback of generating an HDRI from a set of LDRIs is that the total capture time with a standard camera is at least the sum of the exposure times used for programmable cameras. It can increase further for non-programmable cameras as the user needs to change the exposure setting manually between the captures. However, in between each LDRI capture, the environment can change or the camera can move. This is especially true for uncontrollable, outdoor scenes and when not using a tripod. In such cases, combining LDRIs results in an incorrect radiance reconstruction when using the currently available HDRI generation tools, such as HDRshop [6], Rascal [14] or Photomatix [17].

The method presented in this paper, takes as input a set of manually captured LDRIs and allows the following two types of movement to take place during the LDRI capture:

- **Camera Movement**: While taking LDRIs, the camera can move due to lens focusing, user movement, or the user being on a moving platform such as a boat. The presented method allows the LDRIs being captured with a hand-held camera.
- **Object Movement**: During the LDRI capture objects are allowed to move between different frames. The movement does not need to be of a high contrast nature. The only restriction imposed is that the moving object is reasonably small, in order not to interfere with the camera

[1]VECG, University College London, UK
[2]GGG, Universitat de Girona, Spain
[3]Anyhere Software, USA

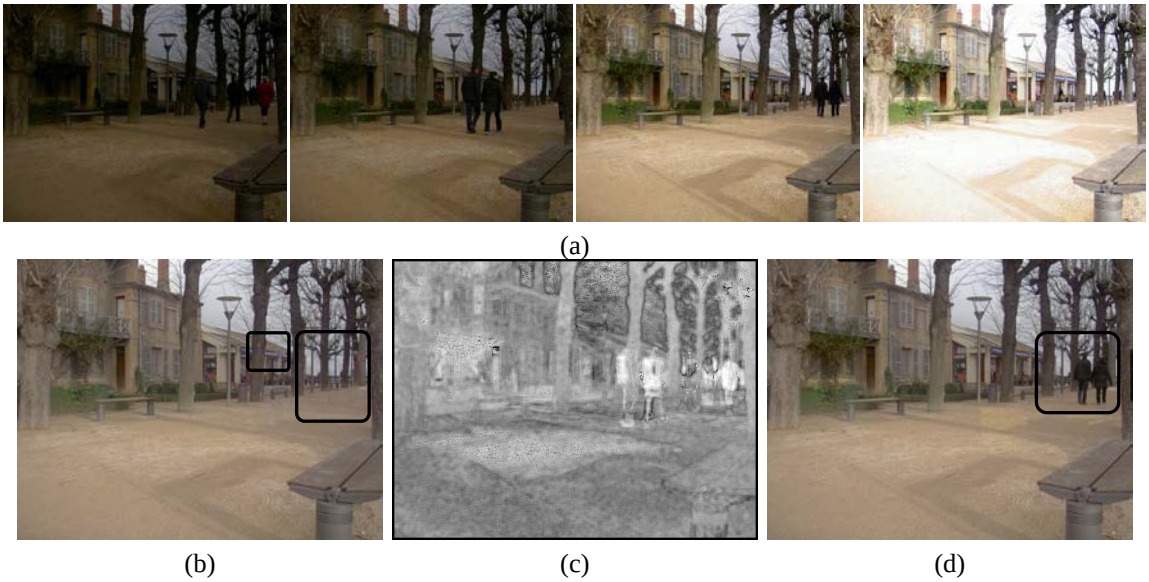

Fig. 1: (a) A sequence of LDRIs captured with different exposure times. Several people walk through the viewing window. (b) An HDRI created from the sequence shown in (a) using conventional methods, showing ghosting effects (black squares). (c) Uncertainty image (UI) shows regions of high uncertainty (bright) due to the dynamic behaviour of those pixels. (d) HDRI of the same scene after applying movement removal using UI.

alignment, and that the area affected by the moving object is captured without saturation or under-exposure in at least one LDRI.

In this paper, a fully automatic framework is presented that aligns LDRIs and combines them while removing the influence of object movement in the final HDRI. Moving objects are automatically identified using statistical quantities and reconstructed from one LDRI during the HDRI generation. The resultant HDRI is free from visible artifacts.

An example is illustrated in figure 1. The sequence of LDRIs shown in (a) shows several persons walking through the viewing window of the camera. In (b) an HDRI is shown that shows ghosting effects inside the black square due to the object movement visible in (a). In (c) an uncertainty image (UI) is shown that defines regions of uncertainty about the static behaviour of the pixels in that area. UI is created using local entry differences between the LDRIs. Using this uncertainty image UI, the movement areas in (b) are substituted with HDR information from one careful selected LDRI. The resulting HDRI shown in (d) is now free from artefacts.

The remainder of this paper is organised as follows. Section II gives an overview of the related work. An overview of the presented system is given in section III. Subsequently the algorithms for the camera alignment, movement detection, and the HDRI generation are explained in sections IV, V and VI respectively. The results obtained are discussed in section VII. Finally a conclusion and some future work are given in section VIII.

II. Background

Conventional HDRI generation methods using multiple exposures [12][4] depend on a good alignment between the LDRIs. Usually they require the use of a tripod throughout the capture, some provide a manual image alignment tool such as provided in the Rascal suite [18]. The larger context of image registration and alignment is well-studied in the computer vision community. For a good survey see [3]. However, few of these methods are robust in the presence of large exposure changes. This presents a particular challenge for automatic

alignment algorithms in cases where the camera response function is not known a priori, since the response curve cannot be used to normalize the LDRIs in a way that would make feature detection and matching reliable.

Four solutions have been presented for image alignment in an HDRI building context. Ward [23] introduced the median threshold bitmap (MTB) technique, which is insensitive to camera response and exposure changes, demonstrating robust translational alignment. Bitmap methods such as MTB are fast, but ill-suited to generating the dense optical flow fields employed in local image registration. Kang *et al.* [9] presented a method that relies on the camera response function to normalize the LDRIs and perform local image alignment using gradient-based optical flow. Sand and Teller [20] presented a feature-based method, which incorporates a local contrast and brightness normalization method that does not require knowledge of the camera response curve[21]. Their match generation method is robust to changes in exposure and lighting, but faces challenges when few high-contrast features are available, or features are so dense that matches become erratic. This is often the case for natural scenes, whose moving water, clouds, flora and fauna provide few static features to establish even a low-resolution motion field. This is where both papers bring in sophisticated techniques, hierarchical homography in the case of Kang *et al.*, and locally weighted regression in the case of Sand and Teller, to overcome uncertainties in the image flow field. Even so, local image warping becomes less reliable as contrast decreases, leading to loss of detail in regions of the image. Furthermore, moving objects may obscure parts of the scene in some exposures and reveal them in others, leading to the optical flow parallax problem, where there is not enough information at the right exposure to reconstruct a plausible HDRI over the entire image. Very recently, Tomaszewska and Mantiuk [22] have proposed an algorithm to align LDRIs captured with a hand-held camera. The algorithm matches key points found with an automatic algorithm that are then used to find the transformation matrix solving for general planar homography.

A different approach by Khan *et al.* [10] has recently tackled the problem of moving objects, and proposed to remove ghost artefacts by adapting weights to validate each pixels to create the final HDRI. Weights are calculating from the probability of each pixel to be part of the background. The algorithm seems to produce similar results to ours, although the examples shown are composed of simple scenes. There is no evidence yet that it could equally work with low contrast backgrounds like our algorithm does.

The nominal reason for warping pixels locally between the LDRIs is to avoid blurring and ghosting in the HDRI composite. With the presented method the need for image warping is removed by observing that each LDRI is a self-consistent snapshot in time, and in regions where blending images would cause blurring or ghosting due to local motions, an appropriate choice of input LDRI to represent the motion will suffice. This approach allows us to apply robust statistics for determining where and when blending is inadequate, and avoids the need for parallax fill. Certain regions may be slightly noisier than they would be with a full blend, but this is an accustomed form of image degradation, and preferable to the ghosts effects that result from improper warping and parallax errors.

The success of and the need for HDRIs have encouraged the development of cameras with built-in HDRI processing [2][15][16]. Even an extension to MPEG video is under consideration [13]. However, the problem of non-static environments remains. With HDR cameras, the time required to take a picture decreases but always remains greater than the longest exposure time used to capture the set of LDRIs. Many of the methods we describe could also be incorporated in HDRI cameras, to reduce the appearance of artifacts.

III. HDRI GENERATION: AN OVERVIEW

A schematic overview of the general HDRI generation methodology is given in figure 2. A sequence of N LDRIs, labelled L_i, are captured with changing exposure settings. Small misalignments might exist between these L_i's. In the presented method, these are approximated by rotational and

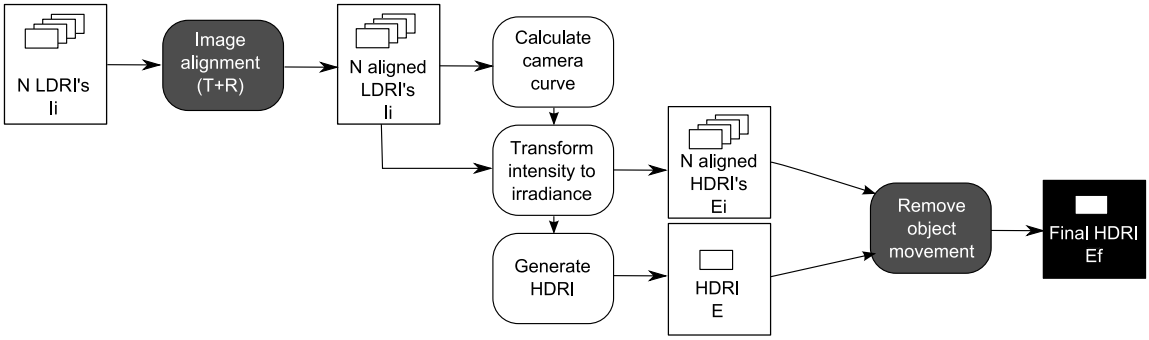

Fig. 2: HDRI generation methodology: the rounded white and grey boxes are processes that operate on input data and produce output data. The rounded grey boxes are modules developed for this paper.

translational misalignments around the viewing direction which are recovered using the method presented in section IV. After alignment the L_i's are used to calculate the camera response curve. The camera curve is used to map the intensity values in L_i to irradiance values, creating a set of N floating point images, labelled E_i. To generate the final HDRI E_f, first the HDRI E is generated in the conventional manner. Then the irradiance values in regions containing object movement are removed and substituted by irradiance information from one E_i. Note that this overview makes abstraction of how and when the movement detection proceeds.

IV. Camera alignment

Usually, small camera movements are inevitable throughout the L_i capture, especially when the images are captured without the use of a tripod and/or the exposure settings are set manually. It is usually fair to assume that the camera movements are small compared to the geometric dimensions of the scene being captured. In this paper it is assumed that the transformation can be approximated as a Euclidean transformation (rotation and translation). The presented method is an extension to the alignment provided by Photosphere [1] which, until recently, only recovered camera translations. More information about the camera alignment implemented in Photosphere is given in [19].

Alignment algorithms often use scene features such as edges or pixel intensities to calculate the camera transformations. Detecting similar scene features in the L_i's is error-prone as they often represent different scene content: different intensities, different colours and edges due to under- or over-exposure effects. An example is given in figure 3. In (a) and (b) two LDRIs, captured with a different exposure setting, are shown. Applying a Canny Edge Detector on (a) and (b) results in respectively (c) and (d). The edges of the shadow shown in (a) are clearly not properly detected in (c).

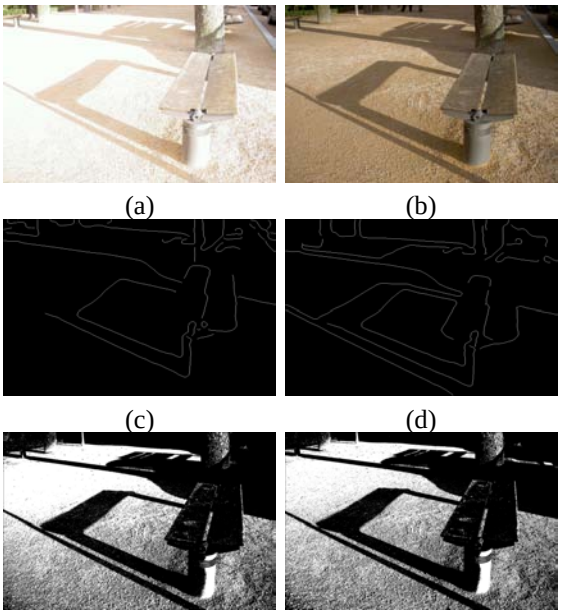

Fig. 3: (a,b) Two LDRIs captured with different exposures. (c,d) Edge images of the two LDRIs. (e,f) Bitmap images of the two LDRIs after applying MTB transformation.

To align the L_i's effectively, the median threshold bitmap (MTB) transform [23] is adopted, which uses the median intensity value (MIV) of an L_i as a threshold to transform that L_i into a binary image $\widehat{L_i}$. MIV splits the pixels in the L_i's into approximately the same two groups, when saturation effects are kept to a minimum. An example of two binary images obtained with the MTB technique is given in figure 3 (e) and (f).

The alignment itself is implemented as an iterative process, where rotational and translational misalignments are minimized until convergence. To speed up this process and to reduce the chance of finding a local minima, the search is implemented on a binary image tree. The alignment will make use of the MTB technique to align two exposures. The XOR difference between two binary images $\widehat{L_i}$ and $\widehat{L_j}$ obtained after applying the MTB transform on two exposures L_i and L_j gives a measure for error. Similarly to [23], the alignment procedure will find the best transformation $T(\cdot)$, consisting of a translation vector $[T_x, T_y]$ and rotation angle α around the center of the image, that when applied to L_i results in the maximum correlation between the two binary images $\widehat{T(L_i)}$ and $\widehat{L_j}$.

The alignment of a sequence of LDRIs is implemented as follows. The middle exposure is chosen as the ground truth; all other exposures are aligned with respect to this exposure. The middle exposure L_m or at least the exposure captured in the middle of the exposure sequence is in general the best aligned with all other exposure in the sequence. Each exposure L_i ($i \neq m$) is aligned with the middle exposure L_m, using a binary image tree, similar to described in [23]. The binary image tree of size Λ ($\Lambda = 4$ in our case) is constructed as follows. The original images $L_i = L_i^0$ and $L_m = L_m^0$ reside at the lowest level ($\lambda = 0$). At the other levels $\lambda \in [1, \Lambda]$, the images L_i^λ and L_m^λ are down-sampled versions of the original images with a down-sample factor equal to 2^λ. The images L_i^λ ($i \neq m$) and L_m^λ are first aligned at level $\lambda = \Lambda$. The calculated transformation is used as a start seed at level $\lambda = \Lambda - 1$, where a new transformation matrix is calculated based on the images with down-sample factor $2^{\Lambda-1}$. This process is repeated until $\lambda = 0$. At a certain level λ the best transformation $T(\cdot)$ (rotation and translation combined) returns the minimum difference between the binary images resulting from applying the MTB procedure on L_m^λ and on the transformed image $T(L_i^\lambda)$. The optimal transformation $T(\cdot)$ is found as the minimum of a set possible transformations. First the optimal translation $[T_x, T_y]$ (in steps of one pixel) is found, followed by the best rotation α (in steps of 0.5 degrees), and this process is iterated until the error converges. The search for this minimum can fail due to local minimum, but is less likely to get stuck in a local minima than when no binary tree is used.

The stability of the MTB alignment method suffers from noisy pixel intensities around MIV, which have an undefined influence on the binary threshold image. This instability can effectively be controlled by withholding the noisy pixel intensities from the alignment procedure, i.e., by excluding pixel intensities that lie within a certain range of MIV. Alignment is achieved as long as moving objects are small compared to the dimensions of the scene, or as long as these moving objects do not create features in the binary images $\widehat{L_i}$. The obtained Euclidean transformation will not be equal to the exact camera transformation, therefore small misalignments may still be present.

V. Movement detection

The movement detection phase, will detect *movement clusters* which are clusters of pixels that are affected by movement in any of the LDRIs. During the HDRI generation, these movement clusters will be analyzed and used to remove the ghosting effects, as will be explained in section VI.

Photosphere [1], and also explained in [19], offers a manner to detect movement clusters using a variance measure. While the method offers good results in most LDRI sequences corrupted by movement, it has the disadvantage that it requires the camera curve to be known and that it relies on high contrast between moving object and background. Section V-A gives details about the variance detector, and specifies when such a method will fail to detect movement. Based on these findings, we decided to

develop a new type of movement detector based on the concept of entropy. The advantage of this method is that it does not require the knowledge of the camera curve, and that it is independent of the contrast between moving object and background. The resulting contrast-independent movement detector is explained in section V-B.

A. Movement detection based on variance

The pixels affected by movement, show a large irradiance variation over the different E_i's. Therefore, the variance of a pixel over the different E_i's can be used as a likelihood measure for movement. The movement cluster is derived from a *Variance Image (VI)*, which is created by storing the variance of a pixel over the different exposures in a matrix with the same dimensions as the images L_i and E.

Some pixels in the L_i's will be saturated, others can be under-exposed. Such pixels do not contain any reliable irradiance information, compared to their counterparts in the other exposures. When calculating the variance of a pixel over a set of images, it is important to ignore the variance introduced by saturated or under-exposed pixels. This can be achieved by calculating the variance $VI(.)$ of a pixel (k,l) as a weighted variance described in [19] as:

$$VI(k,l) = \frac{\sum_{i=0}^{N} W_i(k,l)E_i(k,l)^2 / \sum_{i=0}^{N} W_i(k,l)}{(\sum_{i=0}^{N} W_i(k,l)E_i(k,l))^2 / (\sum_{i=0}^{N} W_i(k,l))^2} - 1 \quad (1)$$

The weights $W_i(k,l)$ are the same as those used during the HDRI generation. The variance image can be calculated for one colour channel or as the maximum of the variance over three colour channels.

The movement clusters are now defined by applying a threshold T_{VI} on VI, resulting in a binary image VI_T. This will not result in nice, well-defined, closed areas of movement clusters due to outliers (false positives and false negatives). For instance, after having aligned the LDRIs with the method described in section IV some camera misalignments might remain. This will result in high variant pixels in VI (especially in the vicinity of edges) that are not due to object movement. To define well-defined, closed, movement clusters, the morphological operations erosion and dilation are applied to the binary image VI_T. A suitable threshold T_{VI} is 0.18.

The generation of the variance image makes use of the irradiance values of the pixels in the E_i's and therefore the variance image generation can only proceed after the camera curve calibration. The incorporation of the movement detection in the general HDRI generation framework, previously shown in figure 2, is given in figure 4.

The method presented so far, defines that high variant pixels in VI indicate movement. It is important to investigate what other influences exist, besides remaining camera misalignments, that might result in a high variant VI value:

- Camera curve: the camera curve might fail to convert the intensity values to irradiance values correctly. This influences the variance between corresponding pixels in the LDRIs and might compromise the applicability of the threshold to retrieve movement clusters.
- Weighting factors: saturation and under-exposure of pixels in an LDRI can result in incorrect irradiance values after transformation to irradiance values using the camera curve. Defining the weighting factors is not straightforward and various different methods exist to define the weights [19].
- Inaccuracies in exposure speed and aperture width used: in combination with the camera curve this produces incorrect irradiance values after transformation. Changing the aperture width causes the depth-to-field to change too, which influences the quality of the irradiance values.

Relying on the fact that the camera curve transforms correctly the intensity images L_i to irradiance images E_i can be seen as a limitation of the variance detector. Though it is true that if the camera curve does not transform correctly the intensities to irradiance values the HDRIs do not represent correctly the environment, there still might be applications

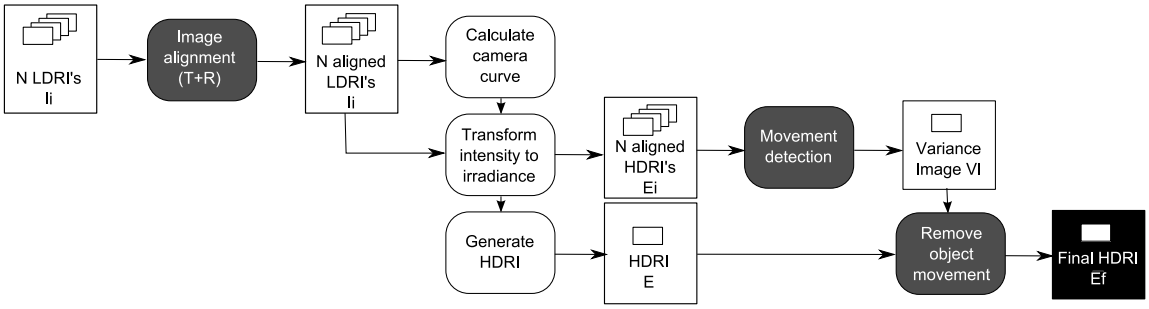

Fig. 4: An adaptation of figure 2 illustrates where the movement detector based on variance fits inside the general HDRI generation framework. The variance detector requires the knowledge of the camera curve, and therefore the movement detector takes place after the camera curve calibration.

for which small errors in HDR values might not be disastrous while it is important to remove the ghosting effects.

The following section presents a method to detect movement without requiring the camera curve. If the camera curve calibration can occur after the movement detection, the detected movement clusters could potentially be used throughout the camera curve calibration to indicate corrupted image data. This will improve the camera curve calibration.

B. Contrast-independent movement detection

In this section we will describe a method to detect movement clusters in an image using a statistical, contrast-independent measure based on the concept of entropy.

In information theory, entropy is a scalar statistical measure defined for a statistical process. It defines the uncertainty that remains about a system, after having taken into account the observable properties. Let X be a random variable with probability function $p(x) = P(X = x)$, where x ranges over a certain interval. The entropy $H(X)$ of a variable X is given by:

$$H(X) = -\sum_{x} P(X = x) \log(P(X = x)). \quad (2)$$

To derive the entropy of an image L, written as $H(L)$, we consider the intensity of a pixel in an image as a statistical process. In other words, X is the intensity value of a pixel, and $p(x) = P(X = x)$ is the probability that a pixel has intensity x. The probability function $p(x) = P(X = x)$ is the normalized histogram of the image. Normalized means that the sum of the probabilities needs to be one. Therefore we divide the histogram values by the total number of pixels in the image. The pixel intensities range over a discrete interval, usually defined as the integers in $[0, 255]$, but the *number of bins* M of the histogram used to calculate the entropy can be less than 256.

The entropy of an image provides some useful information about that image. The following remarks can be made:

- The entropy of an image has a positive value between $[0, \log(M)]$. The lower the entropy, the less different intensity values are present in the image; the higher the entropy, the more different intensity values there are in the image. However, the actual intensity values do not have an influence on the entropy.
- The actual *order* or *organization* of the pixel intensities in an image does not influence the entropy. As an example, two images with equal amounts of black and white intensity values have the same entropy, even if in the first image black occupies the right side of the image and white the left side, and in the second image black and white are randomly distributed.
- Applying a scaling factor on the intensity values of an image does not change its entropy, if the intensity values do not saturate. In fact, the entropy of an image does not change if an *injective function* is applied to the intensity

values. An injective function associates distinct arguments to distinct values, examples are the logarithm, exponential, scaling, etc.

- The entropy of an image gives a measure of the uncertainty of the pixels in the image. If all intensity values are equal, the entropy is zero and there is no uncertainty about the intensity value a randomly chosen pixel can have. If all intensity values are different, the entropy is high and there is a lot of uncertainty about the intensity value of any particular pixel.

The movement detection method discussed in this section has some resemblance to that presented in [11] and [8]. Both methods detect movement in a sequence of images, but restrict this sequence to be captured under the same conditions (illumination and exposure settings). Our method can be used to a sequence of images captured under different exposure settings. Our method creates an uncertainty image UI, which has a similar interpretation as VI. Pixels with a high UI entry indicate movement. The following paragraphs explain how the calculation of UI proceeds.

For each pixel with coordinates (k, l) in each image L_i the local entropy is calculated from the histograms constructed from the pixels that fall within a 2D window W with size $(2w+1)\times(2w+1)$ around (k, l). Each image L_i therefore defines an *entropy image* H_i, where the pixel value $H_i(k, l)$ is calculated as:

$$H_i(k,l) = -\sum_{x=0}^{M-1} P(X = x)\log(P(X = x)) \quad (3)$$

where the probability function $P(X = x)$ is derived from the normalized histogram constructed from the intensity values of the pixels within the 2D window W, or over all pixels p in:

$$\{p \in L_i(k - w : k + w, l - w : l + w)\} \quad (4)$$

From these entropy images a final *Uncertainty Image* UI is defined as the local weighted entropy difference:

$$UI(k,l) = \sum_{i=0}^{N-1}\sum_{j=0}^{j<i} \frac{v_{ij}}{\sum_{i=0}^{N-1}\sum_{j=0}^{j<i} v_{ij}} h_{ij}(k,l) \quad (5)$$

$$h_{ij}(k,l) = |H_i(k,l) - H_j(k,l)| \quad (6)$$

$$v_{ij} = min(W_i(k,l), W_j(k,l)) \quad (7)$$

It is important that the weights $W_i(k, l)$ and $W_j(k, l)$ remove any form of under-exposure or saturation to ensure the transformation between the different exposures is an injective function. Therefore they are slightly different from those used during the HDRI generation. We used a relatively small hat function with lower and upper thresholds equal to 0.05 and 0.95 for normalized pixel intensities. The weight v_{ij} is created as the minimum of $W_i(k, l)$ and $W_j(k, l)$, which further reflects the idea that under-exposed and saturate pixels do not yield any entropic information.

The reasoning behind this *uncertainty measure* follows from the edge enhancement that the entropy images H_i provide. The local entropy is high in areas with many details, such as edges. These high entropic areas do not change between the images in the exposure sequence, except when corrupted by a moving object or saturation. The difference between the entropy images therefore provides a measure for the difference in features, such as intensity edges, between the exposures. Entropy does this without the need to search for edges and corners in an image which can be difficult in low contrast areas. In fact, the entropy images are invariant to the local contrast in the areas around these features. If two image regions share the exact same structure, but with a different intensity, the local entropy images will fail to detect this change. This can be considered a drawback of the entropic movement detector as it also implies that when one homogenous coloured object moves against another homogeneously coloured object, the uncertainty measure would only detect the boundaries of the moving objects of having changed. Nevertheless, real-world objects usually show some spatial variety, which is sufficient for the uncertainty detector to detect movement. Therefore the indifference to local contrast is only an advantage, in particular compared to the variance detector discussed in section V-A.

The difference in local entropy between two images induced by the moving object, depends on the difference in entropy of the moving object and the

background environment. Though the uncertainty measure is invariant to the contrast of these two, it is not invariant to the entropic similarity of the two. For instance, if the local window is relatively large, the moving object is small relative to this window, and the background consists of many similar static smaller objects, then the entropic difference defined in equation 5 might not be large. Decreasing the size of the local window will result in an increased entropic difference, but a too small local window might be subject to noise and outliers. In the current implementation a window size of 5×5 returned good results.

Similarly to section V-A, the movement clusters are now defined by applying a threshold T_{UI} on UI, resulting in a binary image UI_T. Again, this will not result in nice, well-defined, closed areas of movement clusters due to outliers (false positives and false negatives). To define well-defined, closed, movement clusters, the morphological operations erosion and dilation are applied to the binary image UI_T. A threshold T_{UI} equal to 0.7 for $M = 200$ returned satisfactory results. It should be noted though that this threshold did not seem to be as robust as the threshold for the variance detector.

Figure 5 illustrates the movement detection using the uncertainty image UI within the general framework given in figure 2. The creation of UI is independent from the camera curve calibration. As mentioned earlier, this has as an extra advantage that the detection of movement clusters could potentially be used during the camera calibration phase.

VI. HDRI GENERATION

To generate E_f the intensity values in L_i are mapped to irradiance values in E_i, and the HDRI E is constructed in a conventional manner, as a weighted average of the irradiance values in the E_i's. For each movement cluster, the irradiance values in E are substituted by the irradiance values from only one E_i. Similar to [19] for the variance detector, this E_i value is chosen in order to represent the region with the least saturation. When more than one E_i is suitable, the E_i value with the longest exposure time is chosen.

Substituting an entire region with irradiance values from one E_i introduces artifacts at the borders of that region. To reduce these artifacts, only pixels values that have a VI or UI entry above a certain threshold T (higher than the threshold used the movement clusters in the first place) are substituted by a weighted average of the original value in E and the irradiance value in the elected E_i.

VII. RESULTS

In this section some results are shown for the camera alignment, explained in section IV, and the movement detector, explained in section V. All HDRIs shown result from a sequence of LDRIs captured with a hand-held camera and are preceded by a camera alignment prior to the HDRI generation or movement detection, unless stated otherwise.

Figure 6 shows the HDRI generation when no alignment (a,d), translational alignment (b,e) and translational and rotational alignment (c,f) are carried out. The left column shows the entire image, the right column shows an image detail in close-up. The strange blue and pink colours visible in these close-ups are the result of the improper weighting of misaligned pixels during the HDRI generation. In (f), after recovering the translational and rotational transformation, the misalignments are the least visible.

Figure 7 illustrates the performance of the variance and uncertainty detector applied to the exposure sequence shown in figure 1 (a). (a) illustrates the HDRI after movement removal using the variance image shown in (c). (b) illustrates the HDRI after movement removal using the uncertainty image shown in (d). As expected (a) and (b) are similar, which indicates that VI and UI detect the same movement clusters.

Figure 8 illustrates similar results. (a) illustrates the HDRI without any object movement removal. The leaves on the foreground and on the right hand side show considerable object movement. (b) illustrates the HDRI after movement removal using the uncertainty image UI shown in (d). For comparison the variance image VI is given in (c). VI and UI have high (bright) values for the borders of the leaves in the foreground.

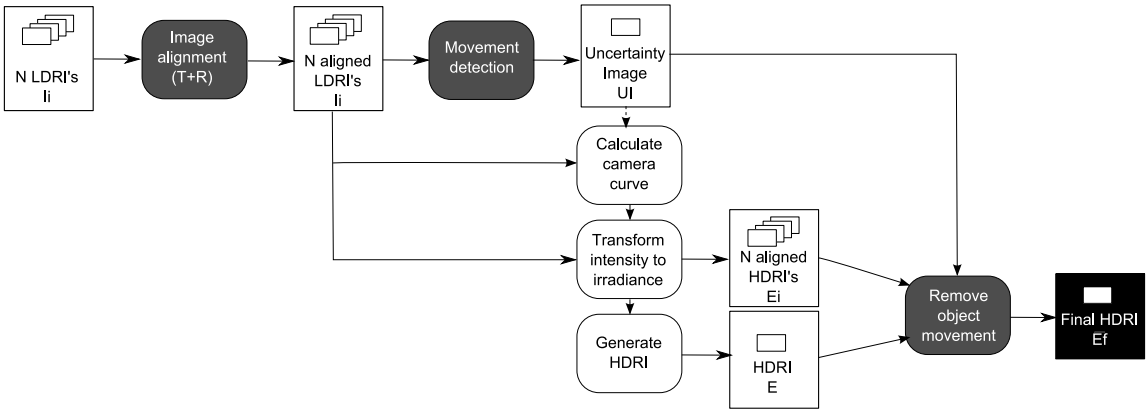

Fig. 5: An adaptation of figure 2 illustrates where the contrast-independent movement detector, explained in section V-B, fits inside the general HDRI generation framework. The movement detector does not require the knowledge about the camera curve, therefore the movement detector can take place before the camera calibration.

The following example illustrates the power of the uncertainty image to detect movement regardless of the contrast between moving objects and the background. Figure 9 (a) shows three LDRIs; the tree branches and the leaves move throughout the capture. The variance image (c) detects the movement around the border of the tree correctly but with more strength than movement of the branches inside the tree. The uncertainty image (b) detects movement inside the tree and near the border with a similar strength, but fails to make a judgement about the sky due to too many saturation and under-exposure effects in that area (only the first exposure shows that are without saturation). The HDRI before and after movement removal using the uncertainty image are shown in figure (d) and (e).

VIII. Conclusion and future work

A method is presented to create an HDRI from a set of LDRIs captured with different exposure settings. During the LDRI capture some camera and object movements are allowed. The potentially negative influences of these movements are effectively removed with algorithms presented in sections IV, V and VI. These algorithms do not require user input, and those compensating for object movements rely on statistical measurements. The final HDRI is free form visible artifacts, although it should be noted that the final HDRI is only an approximation of the scene's true irradiance values.

Though the presented method is reasonably robust and takes care of some camera and object movement, some caution is needed. There are still a few possible scenarios for which HDRI generation remains error-prone.

When a too large object (an object occupying a large area in the LDRI) moves in the scene, the presented alignment procedure may fail to align the different LDRIs. Even when alignment is successful the camera curve reconstruction will be erroneous using the conventional camera calibration algorithms, and the resulting HDRI will be incorrect nonetheless. However, this paper presents a movement detection method, independent from the camera curve, that returns movement clusters that could be used during the camera curve calibration, which is left as future work.

In the final HDRI movement clusters are substituted by irradiance values from that LDRI that does not show saturation in that particular area. It is possible, however, that no suitable LDRI is available and as a result the irradiance values in those regions contain incorrect irradiance information.

Besides camera and object movement, it is possible that during the LDRI capture the scene illumination changes as well, for instance due to cloud movement. This has a significant impact on the

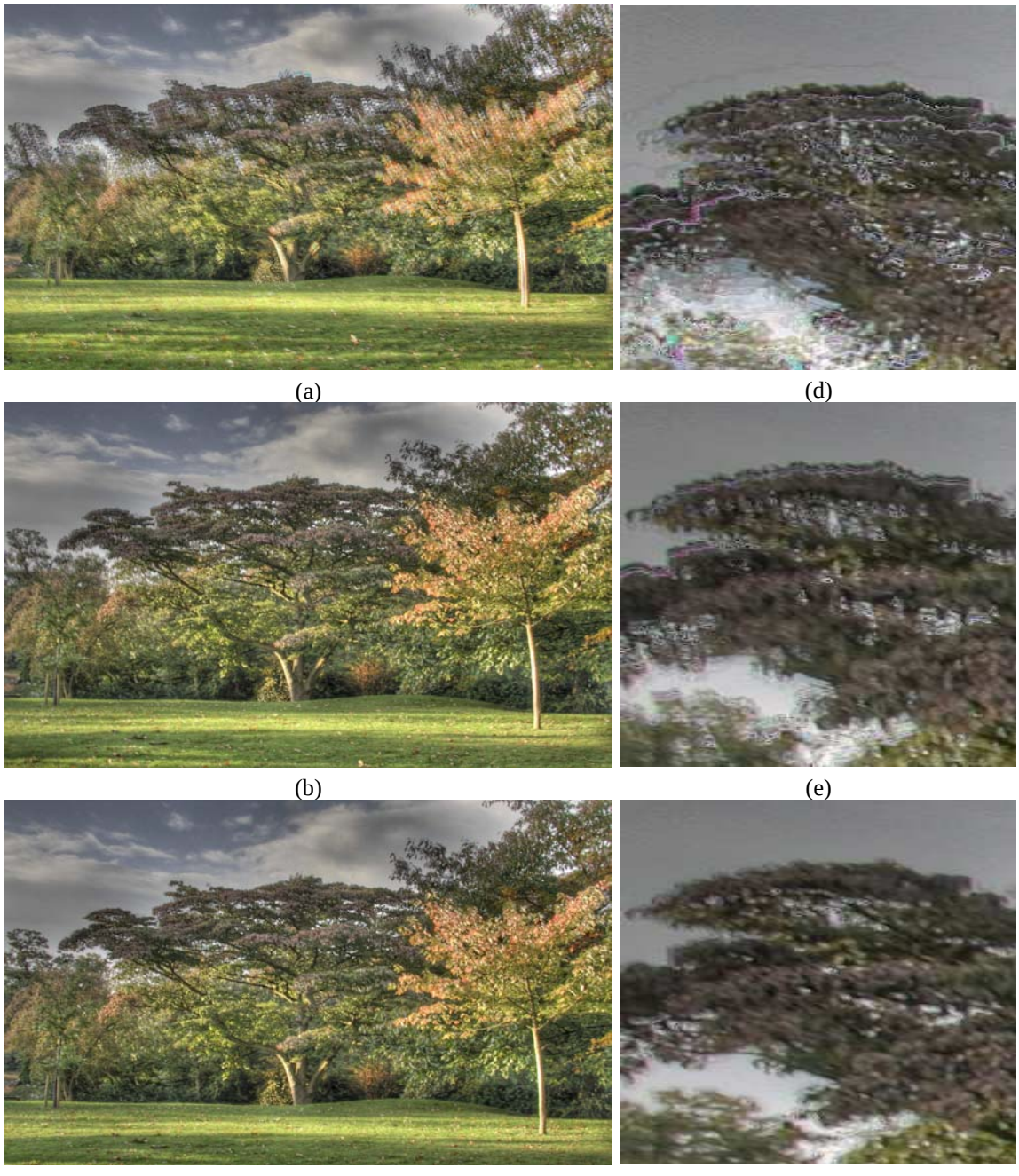

Fig. 6: HDRI generation and the influence of camera movement. The left column shows the entire HDRI, the right column shows an image detail in close-up for the following scenarios: no image alignment (a,d), translational alignment (b,e), translational and rotational alignment (c,f).

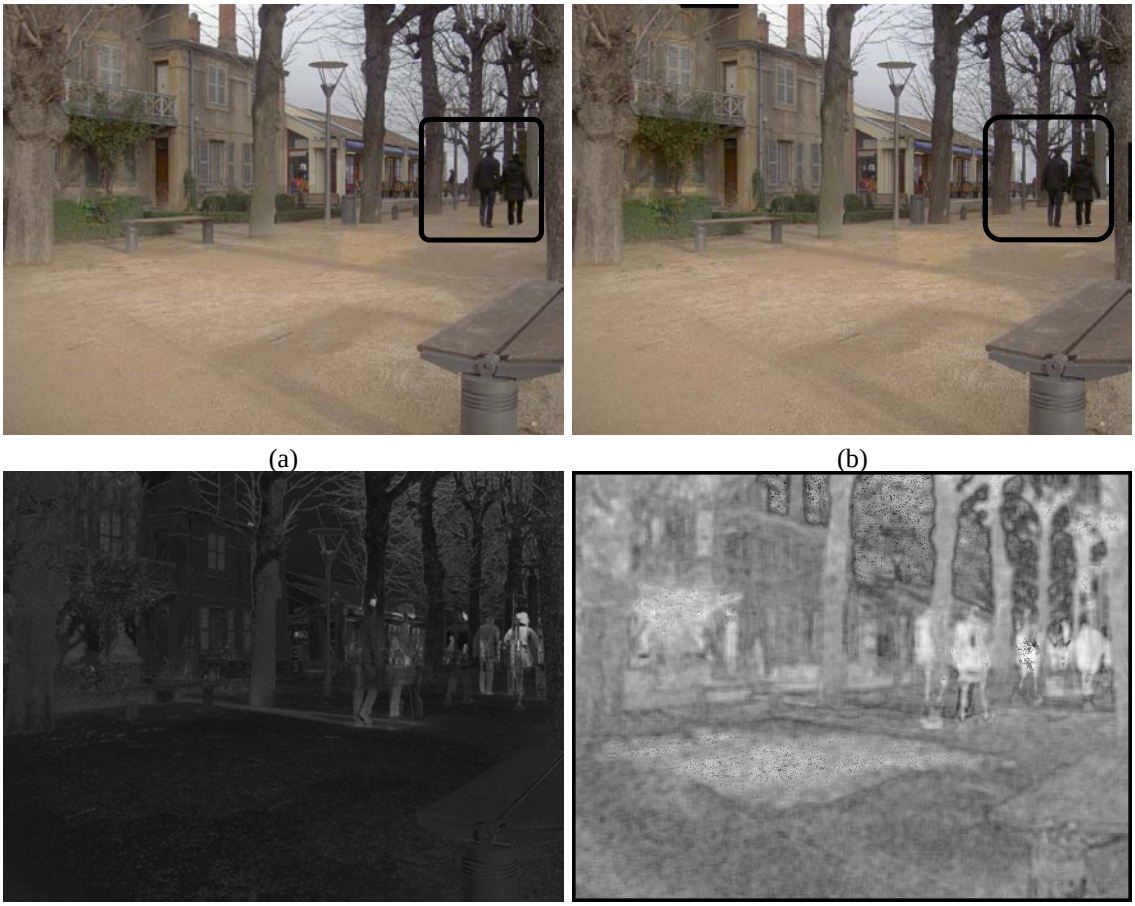

Fig. 7: HDRI generation and movement removal for the exposure sequence shown in figure 1 (a). (a) HDRI after object movement removal using the variance detector discussed in section V-A. (b) HDRI after object removal using uncertainty detector discussed in section V-B. (c) The variance image VI used to generate (a). (d) The uncertainty image UI used to generate (b).

HDRI generation and so far no solutions have been proposed to take care of these illumination changes.

The uncertainty image offers advantages compared to the variance image. Firstly, it can be generated prior to the camera curve calibration, and secondly, it is contrast-independent. The drawback is that its generation is computationally expensive and that it can fail to make a decision for object with very bright or vary dark irradiance values.

REFERENCES

[1] Anyhere software. Photosphere. http://www.anyhere.com.

[2] BASLER Vision Technology. Basler a600. www.baslerweb.com/beitraege/beitrag_en_17681.html.

[3] Lisa Gottesfeld Brown. A survey of image registration techniques. *ACM Computing Survey*, 24(4):325–376, 1992.

[4] Paul E. Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. In *proceedings of ACM Siggraph '97 (Computer Graphics)*, volume 31 of *Annual Conference Series*, pages 369–378, 1997.

[5] Paul E. Debevec, Camillo J. Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *proceedings of ACM Siggraph '96 (Computer Graphics)*, volume 30 of *Annual Conference Series*, pages 11–20, 1996.

[6] HDRshop. www.HDRshop.com.

[7] Katrien Jacobs and Céline Loscos. Classification of illu-

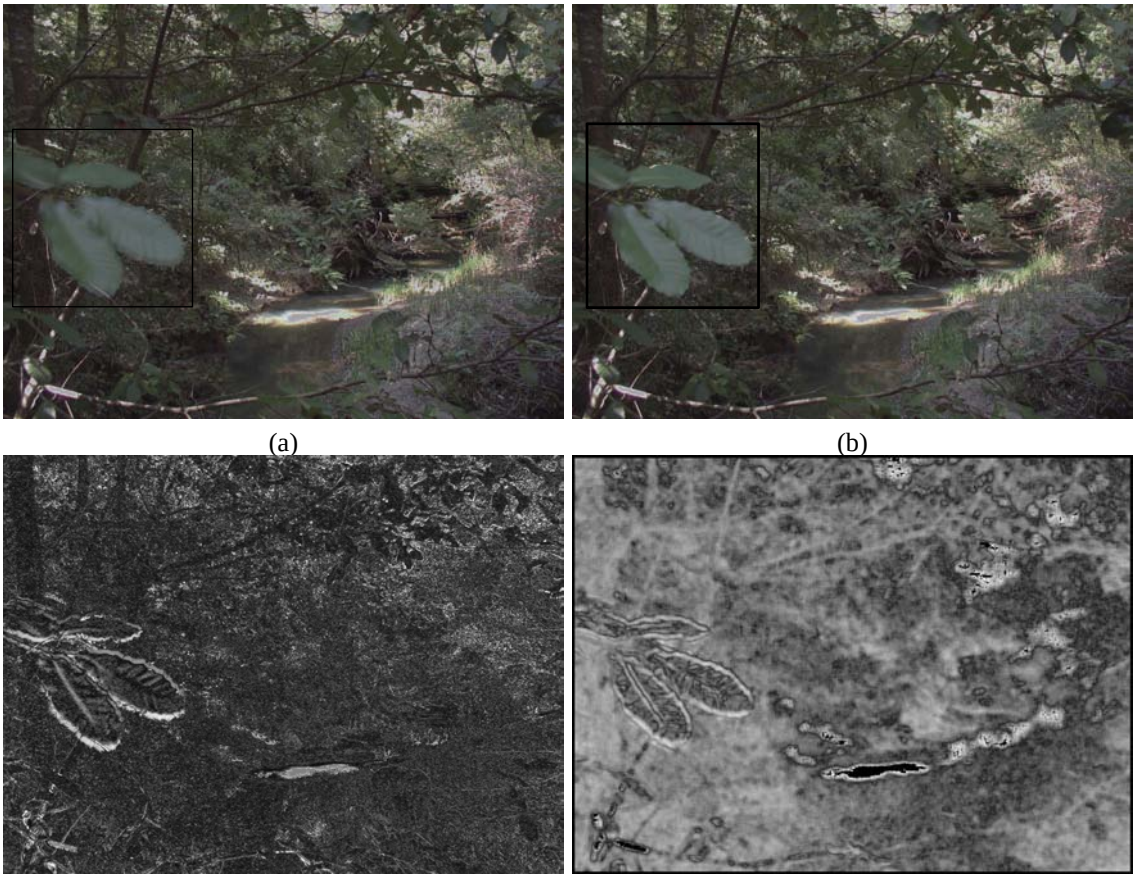

Fig. 8: (a) HDRI without movement removal: the leaves on the left hand side show considerable ghosting. (b) HDRI after movement removal using the uncertainty image UI shown in (d). (c) The variance image VI. (d) The uncertainty image UI used to generate (b).

mination methods for mixed reality. In *State of the Art Report at Eurographics '04*, 2004.

[8] Guo Jing, Chng Eng Siong, and Deepu Rajan. Foreground motion detection by difference-based spatial temporal entropy image. In *TENCON 2004. 2004 IEEE Region 10 Conference*, volume 1, pages 379–382, November.

[9] Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski. High dynamic range video. In *proceedings of ACM Siggraph '03 (Computer Graphics)*, pages 319–325. ACM Press, 2003.

[10] Erum Khan, Oguz Akyuz, and Erik Reinhard. Ghost removal in high dynamic range images. In *International conference on image processing (ICIP)*, 2006. poster.

[11] Yu-Fei Ma and Hong-Jiang Zhang. Detecting motion object by spatiotemporal entropy. In *IEEE International Conference on Multimedia and Expo*, August 2001.

[12] Steve Mann and Rosalind W Picard. Being undigital with digital cameras: Extending dynamic range by combining differently exposed pictures. In *Proceedings of IST 46th annual conference*, pages 422–428, May 1995.

[13] Rafal Mantiuk, Grzegorz Krawczyk, Karol Myszkowski, and Hans-Peter Seidel. Perception-motivated high dynamic range video encoding. In *proceedings of ACM Siggraph '04 (Computer Graphics)*, pages 733–741. ACM press, 2004.

[14] Tomoo Mitsunaga and Shree K. Nayar. Radiometric self calibration. In *proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 374–380. Fort Collins, June 1999.

[15] Shree K. Nayar and Vlad Branzoi. Adaptive dynamic range imaging: Optical control of pixel exposures over space and time. In *proceedings of International Conference on Computer Vision (ICCV)*, 2003.

[16] Shree K. Nayar, Vlad Branzoi, and Terry Boult. Programmable imaging using a digital micromirror array. In

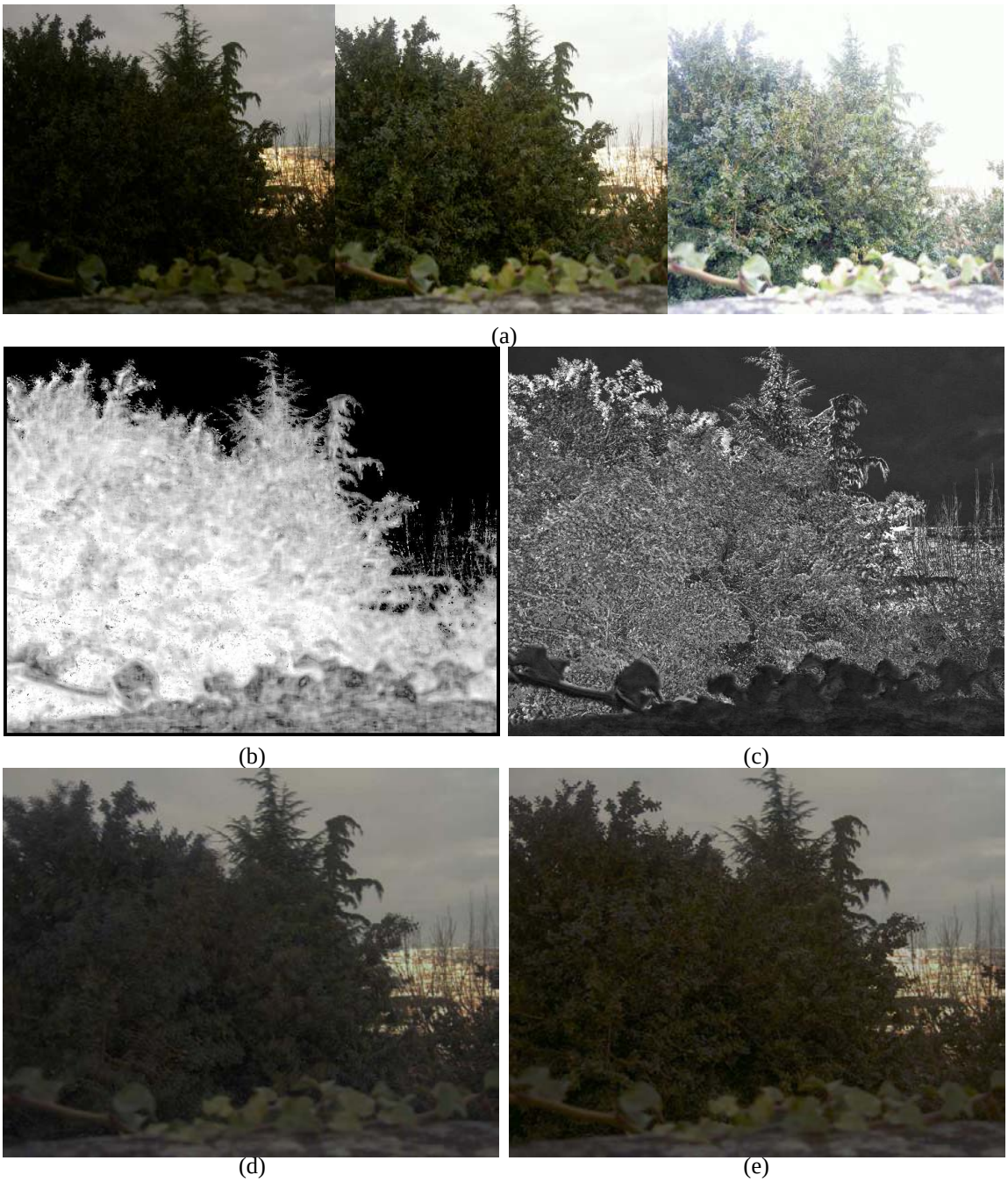

Fig. 9: (a) Image exposure sequence: the branches and leaves in the tree move due to the wind. (b) Uncertainty image UI. (c) Variance Image VI. The resulting HDRI prior to movement removal is shown in (d). The HDRI after movement removal using UI is shown in (e).

proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2004.

[17] Photomatix. Multimediaphoto. www.hdrsoft.com.

[18] Rascal. Radiometric self calibration. www1.cs.columbia.edu/CAVE/tomoo/RRHomePage/rrhome.html.

[19] Erik Reinhard, Greg Ward, Sumanta Pattanaik, and Paul Debevec. *High Dynamic Range Imaging: Acquisition, Display and Image-Based Lighting*. Morgan Kaufmann Publishers, 2005.

[20] Peter Sand and Seth Teller. Video matching. In *ACM Transactions on Graphics (Proc. of Siggraph '04)*, volume 22, pages 592–599, New York, NY, USA, july 2004. ACM Press.

[21] Peter Sand and Seth Teller. Video matching. Technical report, MIT, 2004.

[22] Anna Tomaszewska and Radoslaw Mantiuk. Image registration for multi-exposure high dynamic range image acquisition. *WSCG*, January 2007.

[23] Greg Ward. Fast, robust image registration for compositing high dynamic range photographs from handheld exposures. *Journal of Graphics Tools*, 8(2):17–30, 2004.